Adaptive Methods for the Structural Optimization of Neural Networks and Their Ensemble for Data Analysis

Vladimir Bukhtoyarov(✉), Vladimir Nelyub, Dmitry Evsyukov,
Sergei Nelyub, and Andrey Gantimurov

Bauman Moscow State Technical University, Moscow, Russia
bukhtoyarov@emtc.ru

Abstract. The article is devoted to studying the effectiveness of ensemble methods of neural network solvers for solving a regression problem. The problem of ensuring the efficiency of using a prediction model based on a set of artificial neural networks, which should provide increased efficiency compared to a single regressor, is considered. One of the requirements to ensure an increase in efficiency is that the solvers in the ensemble must be sufficiently different to ensure that the ensemble model exits the region of the local minimum error of a single solver. For the considered case of constructing neural network ensembles, it is proposed to provide distinction based on the generation of different structures of neural network regressors. To do this, a modification is introduced into the structure of the previously developed probabilistic method for designing neural networks. This modification is based on the use of special coefficients that determine the adaptation of the probability of using various activation functions depending on the presence of those in already formed neural networks in the ensemble. The proposed approach was implemented in a software system and tested using generated datasets and real industrial data sets, described in the article. The results obtained indicate the relatively high efficiency of constructing collective regressors when forming neural networks using the proposed approach while maintaining diversity in conditions of noisy samples.

Keywords: Modeling · Simulation · Artificial Neural Networks · Ensembles · Regression · Structure Shaping · Probabilistic Approach

1 Introduction

Today, the use of artificial neural networks makes it possible to effectively solve problems of data mining, such as modeling, classification, and clustering. In the course of the ongoing research, regression modeling problems were considered, involving the use of artificial neural networks to describe the dependencies between input and output data. There are quite a few applications of artificial neural networks in this focus, including in metallurgy, mechanical engineering and medicine [1–3].

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2024
V. Jordan et al. (Eds.): HPCST 2023, CCIS 1986, pp. 143–157, 2024.
https://doi.org/10.1007/978-3-031-51057-1_11

However, we can observe a constant increase in the complexity of data analysis problems. The technological processes themselves and the models that describe them are becoming more complex. This is due to the fact that there is an increase in the number of controlled process parameters and, accordingly, an expansion of the factor space for searching for effective models. There is also an increase in the amount of data processed to build regression models, and the demands on the accuracy of such models are becoming more and more demanding. All this leads to the fact that the process of searching for methods that would allow the formation of more efficient regression models, including criteria for computational efficiency and accuracy of describing dependencies in data, is constantly ongoing. This is true for both regression neural network models and other types of models.

Among several directions for improving approaches to building regression models (and, in general, classifiers), one can single out ensembling. This approach involves the construction and integration of models and classifiers into groups that process data sets more efficiently than with an individual regressor or classifier [4–6]. A limitation of the approach is the fundamental complexity of the ensemble compared to an individual solver. In this regard, the developed schemes for constructing ensembles should be effective in terms of the use of individual solvers. Some of them, in particular, implement the requirements of maintaining diversity in such an ensemble of solvers.

One of the fundamental works in the field of artificial neural ensembles is [7]. The authors have shown an increase in the efficiency of the recognition system when using several neural network classifiers instead of a single artificial neural network. This approach and its variants were refined and implemented to solve practice-oriented problems, including in the field of recognition, medical and technical diagnostics, processing of seismic monitoring signals and many others [8–11].

Despite the demonstrated possibility of using ensembles for certain practical problems, this approach is associated with an increase in the complexity of the model design stage. This is due to the need to build not one, but several at once, sometimes a significant number of neural network solvers, which are required to be effective jointly, and not separately.

There are two steps required to build the ensemble solver. At the first step, it is necessary to create individual solvers that are sufficiently effective and differ from each other (artificial neural networks are considered within the framework of this study). At the second step, it is necessary to assemble an ensemble scheme for obtaining a general solution from individual solvers, namely, their individual solutions. It appears that increasing the efficiency of solving a problem depends on each of the two steps indicated. Thus, the effectiveness of the ensemble approach is determined by the quality of the solvers in the pool formed at the first step, and the effectiveness of the method for calculating the ensemble solution from individual solutions. [6, 12–14]. Different approaches for performing the above steps can be identified in real-world applications. A complete classification is not the purpose of this review, but to determine the location of the presented study, the following must be indicated. Individual classifiers can be built over the entire dataset, or they can be defined on individual subsets of the original dataset. In this study, we considered the concept of the entire dataset for all individual solvers. The construction of a general solution can be based on simple rules such as weighted summation or median, or it can be produced in the form of a more flexible, potentially nonlinear relationship. As part of this study, we integrated into the proposed approach a scheme for generating a general solution based on genetic symbolic programming.

Further, in Sect. 2, a description is given of the proposed methods for the formation of artificial neural networks for their inclusion in the ensemble and, additionally, the method of forming the general solution of the ensemble based on the predictions of individual neural networks. The proposed methods are based on a generalized principle underlying evolutionary heuristic algorithms. Section 3 presents the results of a numerical experimental study of the proposed approaches and a comparative analysis with several alternative ensemble approaches and individual models. Section 4 contains a general conclusion on the article.

2 Materials and Methods

2.1 Designing the Structure of Neural Networks Using the Probabilistic Approach

In the case of using an ensemble approach, special attention should be paid to the stage of generating individual solvers. Such solvers should not be completely untrained, but should have some basic performance that demonstrates their adaptation using the dataset. On the other hand, they should not be overtrained, since there is a possibility of bringing them to some general local optimum in terms of performance criterion for a specific problem. In this sense, they must be different, covering different points in the solver space. A pool of such preliminary predictions may be intended to be used in whole or in part with subset selection. The scheme for the selection of ensemble members will be considered in the next section. In this section, we will consider the proposed approach to the formation of structures of artificial neural networks for subsequent training.

Since the ensemble approach assumes the need to build a sufficiently large pool of individual solvers with a high degree of confidence, the conditions of real application require automation of this process. This is due to the fact that building even one solver is quite a serious task for practical application on datasets of significantly different types. And here this problem of optimal design of an efficient solver must be solved many times with some restrictions on the complexity and variety of individual solvers. This fully applies to the use of artificial neural networks as individual solvers, which are the focus of this study. The issue of adapting neural network solvers can be most effectively resolved in the case of optimal selection of the structure and further effective training of the corresponding network. It is almost impossible to rationally determine in advance the non-redundant efficient structure of a neural network designed for a huge number of real-life datasets. The requirement of non-redundancy is essential for the ensemble approach, since the use of many redundant individual solvers can lead to a significant increase in the need for computing resources compared to the increase in the quality of the resulting solution. Limits on available computing resources may also be exceeded, which is still typical for a large number of real-world systems.

In order to further describe the method for maintaining the diversity of neural network solvers, we provide a brief description of the previously developed probabilistic method for designing the structures of artificial neural networks. This approach is for automatically designing the structure of neural networks is based on the calculation and use of probability estimates (Formula 1).

$$p_{i,j}^{k}, i = \overline{1, N_l}, j = \overline{1, N_{neuron}}, k = \overline{0, N_F}, \tag{1}$$

where i is the number of the hidden layer of the neural network; j is the number of the neuron on the hidden layer of the network; N_l is the maximum number of hidden layers; N_{neuron} is the maximum number of neurons on the hidden layer; k is an identifier whose value is interpreted as follows:

1. If $k = 0$, then $p_{i,j}^{0}$ is an estimate of the probability that the j-th neuron is absent on the i-th layer of the network;
2. If $k \in [1, N_F]$, then $p_{i,j}^{k}$ is an estimate of the probability that the j-th neuron exists on the i-th layer of the network and its activation function is the activation function with number k from the set of activation functions available to the algorithm. N_F is the power of the set of activation functions that can be used when designing the neural network structure.

The formula for calculating the total probability of the presence or absence of a neuron at a place in the neural network, determined by the number of the layer and the number of the conditional position of the neuron on the layer is as follows:

$$p_{i,j}^{0} + \sum\nolimits_{k=1}^{N_F} p_{i,j}^{k} = 1. \tag{2}$$

For details see [15].

2.2 Providing Diversity for Individual Neural Network Solvers

One of the priorities for building effective ensemble models is to ensure the diversity of individual solvers, ensuring the possibility of their synergistic interaction and not being biased towards the local and insufficient maximum efficiency of a single model. Maintaining diversity can be achieved in a variety of ways. This can be building or training models on different subsets of the original dataset, as is implemented in boosting schemes. Another option is to build models with initially different structures. For artificial neural networks, the focus of this research, this means designing and further training neural network solvers with different activation functions and hidden layer configurations.

Within the framework of this study, just such an approach was considered with a focus on the formation of an ensemble based on artificial neural networks. The modification of the method of probabilistic design of neural network structures is as follows. An additional coefficient is introduced that is used to correct the probability of forming the structure of a neural network with each type of activation function used. Obviously, maintaining diversity among neural networks is also ensured by their structural differences. Correcting the probabilities of the procedure producing neural networks should reduce the likelihood of the appearance of neural networks with a structure similar to those that have already been formed and placed in the pool of solvers. It is proposed to use modified probability values calculated according to Formula 3:

$$\tilde{p}_{i,j}^{k} = d_i^k \cdot p_{i,j}^{k}, \tag{3}$$

where $p^k_{i,j}$ i and iterators are defined by Formula (1) The value of the coefficient is calculated by Formula 4:

$$d^k_{i,j} = 1 - \frac{n^k_{i,j}}{Ensemble\ Size}, \tag{4}$$

where n is the number of neural networks, with a neuron of the k-th type on the i-th layer in the j-th position, already placed in the ensemble. The coefficient of change in the probability of occurrence of an activation function of a certain type d^k_i is calculated individually for all neurons of the hidden layers of the designed neural network. Possible values of such coefficients are in the range (0; 1]. If there is no neural network with the corresponding activation function of the neuron in the position of the hidden layer in the generated set of individual solvers, the coefficient is equal to 1. When such neural networks of networks appear, the coefficient decreases and, accordingly, the probability of generating a network with the same neuron in the neuron position in the hidden layers corresponding to the coefficient decreases. This is aimed at reducing the repeatability of the structure of neural networks, which is a factor in maintaining the diversity of individual neural network solvers for the ensemble model. The starting values of the coefficients for constructing the initial neural network are to equate all coefficients to 1.

The approach under consideration in a general formulation was proposed and partly studied, and the results of such a study are given in [16]. However, in the first version of the proposed approach, two important steps were not implemented, which were used and already tested on the new test function number 5, introduced in Sect. 3.

In order to increase the stability of the approach, we introduced and implemented the following modifications, compared to [16]. The formula for calculating the coefficient used to take into account diversity has been changed, which allows us to take into account the number of neurons of a certain type in already formed neural networks $N^k_{i,j}$, , and not in relation to the entire ensemble size (Formula (5)). This is due to the following factor. In the proposed evolutionary and some alternative ensemble approaches the choice of neural regressors directly used in the formation of the ensemble and its general solution occurs at the next step. Therefore, it is not possible to determine the size of the ensemble in advance. The approach used previously involved dividing by the ensemble size (for approaches with a fixed number of solvers in the ensemble) or dividing by the number of networks in the preliminary pool. This gave rise to ambiguity and dependence of the first step of ensemble formation on the second step. The proposed modification eliminates such ambiguity and makes the approach invariant from the choice of approach to forming a general ensemble solution.

$$d^k_{i,j} = 1 - \frac{n^k_{i,j}}{N^k_{i,j}}. \tag{5}$$

In the modified version of the approach proposed here, an unconditional minimum probability was introduced, less than which the probability cannot be decremented using Formula (3). This is due to a motivated relaxation of the calculation rule using decrementation coefficients according to Formula (5) so that it is possible to obtain neural network structures of any configuration at each iteration of designing the preliminary pool or the ensemble. In this study, we used an unconditional minimum value of 0.05 for cases where recalculation of the probability according to Formula (3) leads to a $\tilde{p}_{i,j}^{k}$ value less than the established value of 0.05.

Taking into account the basic statistical conditions, in the new version of the approach to maintaining diversity in ensembles, to ensure the stability of the approach after recalculating the probabilities using Formula (3), a normalization stage has been introduced, bringing the sum of probabilities to a value equal to 1:

$$\tilde{p}_{i,j}^{k} = \frac{\tilde{p}_{i,j}^{k}}{\sum_{k=0}^{N_f} \tilde{p}_{i,j}^{k}}. \tag{6}$$

This step was not introduced and used for the initial version of this approach and had the potential to fail to ensure the completeness of the sum of probabilities, potentially reducing the effectiveness of the approach. Together with minor corrections in iterators, the introduced changes allow us to claim a significant refinement and improvement of the original approach, requiring confirmation using a new test dataset. The improved new approach was also tested using a new, more complex test function with varying levels of imposed noise.

2.3 Ensemble Design Using Modified Genetic Programming

The formation of an ensemble can be carried out in the following options. The first option is that the ensemble consists of all individual solvers. In this case, individual solvers are formed depending on the qualities of previous members of the ensemble or the explicitly or implicitly defined areas of influence of each solver. In particular, this can be attributed to various boosting schemes, which also cover the design stage of individual solvers [10, 17]. An alternative option is to compile an ensemble of solvers selected from a preliminary pool - either a small subset or all of the pool can be selected. An alternative option is to compile an ensemble of solvers selected from a preliminary pool - either a small subset or all of the pool can be selected. Such selection can be carried out in an explicit combinatorial form, which, however, is a computationally intensive process given the large number of combinations with a fairly large size of the designed ensemble. Such selection can also be carried out implicitly, for example, by resetting the weighting coefficients of individual solvers to zero in the formula for calculating the general solution, or even during the construction of the very principle of calculating the general solution, which may not take into account the predictions of all individual solvers. [18–21]. This leads to the fact that one of the problems of effective application of ensemble methods is the choice of a method for forming a general solution. In the basic case, there are two possible approaches. The first one assumes, for the regression problem, the use of some formula that is static. Such methods may include methods for calculating the average value based on the solutions of individual solvers. In addition to simple averaging, a similar scheme with the introduction of weighting coefficients can be used. Such weighting factors may be proportionally related, for example, to an estimate of the error of the corresponding individual solver. However, the issue of choosing coefficients remains, since an individually efficient solver may have too much weight to bias the ensemble solution relative to it, thereby reducing the ensemble effect. An alternative option is to use adaptive schemes for integrating and post-processing the opinions of individual classifiers. In the limit, this can be another level of solvers that accept as input the outputs of individual solvers of the first level. Keeping in mind the possibility of such an approach, the study attempted not to permanently seal the black box of an ensemble of neural networks in this form.

Therefore, we have proposed and tested a method that allows us to adaptively form a single second-level solver in the form of a formula that explicitly symbolically connects the outputs of individual solvers and the general ensemble solution. The scheme for using ensembles of neural networks suggests that the calculation of the general ensemble solution is based on the solutions obtained by individual neural networks, that is, the general ensemble solution is a certain function that depends on the predictions of individual neural networks (5):

$$o = f(o_1, o_2, \ldots, o_n), \tag{3}$$

where o is the general solution, o_i is the individual solution of the i-th network, n is the number of networks in the ensemble. To solve this method of computing an ensemble solution, we propose to use an approach based on genetic programming [22]. To solve this method of computing an ensemble solution, we propose to use an approach based on evolutionary heuristics called genetic programming. It is successfully used to construct optimal approximations, and the explicit formulaic form of the solution will allow one to analyze the composition and evaluate the influence of individual solvers when calculating the overall solution.

The standard genetic programming method is used to design symbolic formulas involving independent variables and constants. Since in the considered ensemble scheme the input values of the generated rule for calculating the general ensemble solution are the output of individual neural network models, a modification of the standard approach is required. For this purpose, a modification of the genetic programming method with a modified terminal set $T = \{o_1, o_2, \ldots, o_n, C\}$ is used. Here o_i is an individual solution of the i-th solver, n is the number of networks in the ensemble, C is a set of constants (numerical coefficients). A hybridized version of the genetic programming method was used in our study to fine-tune the efficiency-critical second stage of ensemble design. Such hybridization consists in the use of algorithms for adjusting the coefficients of a formula that describes a combination of solutions of individual solvers to determine the overall ensemble solution. An important positive aspect of using such evolutionary heuristics as a method of genetic programming is the possibility of flexible formation of a quality function for the process of evolutionary improvement of solutions. So, in the form of a convolution or multi-criteria formulation, such an ensemble efficiency criterion can include not only the values of the model or classifier deviation estimates, but estimates of the number of individual solvers used or their total complexity of neural structures or other types of solvers. This can be critical in cases of limited computing resources and the desire to maintain the speed of inference of the ensemble model compared to individual solvers.

3 Numerical Experiments

To evaluate the effectiveness of the proposed approach, numerical studies were performed on the generated sets of test problems and data sets of real problems hosted in the Machine Learning Repository [23]. The description of the generated and test sets is given below. As a unique objective of this study, a set of data obtained from the metallurgical production at the stage of obtaining end products in plums was used to evaluate the quality of regression modeling. The name of the company that provided the data and the characteristics of the dataset are not disclosed in order to ensure non-disclosure of commercial information.

3.1 Simulated Datasets

For the initial evaluation of the work of the considered methods, the generated data sets were used. The formation of datasets was carried out by randomly generating inputs and calculating the values of functions in the domain of definition and according to the formulas presented in Table 1. In previous studies, the use of such data sets made it possible to assess the adequacy of their use for assessing the effectiveness of regressors with basic indicators of the effectiveness of methods for constructing regression solvers on a number of real data sets. It seems reasonable to us to use such synthetic modeling problems for the purpose of controlled imposition of noise on data to assess the robustness of modeling methods.

3.2 Test Problems

To test the methods considered and proposed in the study for solving regression public datasets hosted in the Machine Learning Repository were also used. We used the Concrete Slump Data Set because it is a widely used publicly available data set that will allow external verification of the results obtained in this study. This data set consists of patterns describing the tensile strength tested for special samples. Such samples are made using concretes with controlled levels of inclusions of various ingredients. The total number of examples in the dataset is 103 observations. A dataset option with an expanded set of observations up to 1030 is also available. In this study, a basic 103 instance dataset was used, directly obtained from field samples.

A data set was also used that describes the implementation of the ore-thermal smelting process with various input parameters characterizing this metallurgical process. The regression modeling problem statement is as follows. Based on observations of a real object, data samples were generated that characterize the efficiency of an ore-thermal smelting furnace. Electrical parameters and the loading of other components are used as control parameters (input influences), since it is these input parameters that influence the processes within the processes, in addition, they provide the ability to continuously obtain and reliable information about them.

As input parameters of the process occurring in the furnace, the following process indicators were recorded: the amount of agglomerate loaded into the furnace; the amount of silica loaded into the furnace; the amount of coke loaded into the furnace; the amount of converter slag loaded into the furnace; electricity input; deepening of electrodes;

Table 1. Functions for generating test data.

Test problem	Simulated Function	Range	Sample size
1	3-d Mexican-Hat: $y = \frac{\sin\sqrt{x_1^2+x_2^2}}{\sqrt{x_1^2+x_2^2}}$	$x_i \in [-2\pi, 2\pi]$	1000
2	Friedman 1: $y = 10\sin(\pi x_1 x_2)+20(x_3-0.5)^2+10x_4+5x_5$	$x_i \in [0, 1]$	1000
3	Friedman 2: $y = \sqrt{x_1^2 + (x_2 x_3 - \frac{1}{x_2 x_4})^2}$	$x_1 \in [0, 100]$ $x_2 \in [40\pi, 560\pi]$ $x_3 \in [0, 1]$ $x_4 \in [1, 11]$	1000
4	Gabor: $y = \frac{\pi}{2} exp\left[-2\left(x_1^2 + x_2^2\right)\right] cos[2\pi(x_1 + x_2)]$	$x_i \in [0, 1]$	1000
5	Multi: $y = 0.79 + 1.27x_1x_2 + 1.56x_1x_3 +$ $3.42x_2x_5 + 2.06x_3x_4x_5$	$x_i \in [0, 1]$	1000

voltage; current strength; specific energy consumption. These parameters at the upper level make it possible to evaluate the energy and technological characteristics of the operating process of such installations and, in general, to judge the efficiency of the furnaces. In order to build a regression model, the nickel content in the waste slag in percent was selected as an output parameter.

3.3 Pool of Methods

The proposed approach was evaluated in terms of the efficiency of building a regression model by comparison with a number of alternative methods. Such alternative methods were implemented in software and used to build models in the considered test problems. The following methods for constructing individual regressors were investigated: artificial neural networks, regression support vector machine, multidimensional adaptive spline method [24, 25].

GASEN method based on the use of a genetic algorithm to select networks from a pool, and a boosting gradient method were considered as alternative ensemble methods [26–28]. Verification of the correctness of the implementation and pre-setting of parameters for the approaches under consideration were performed using this software system during a preliminary study on a basic set of test problems (Table 1).

3.4 Raw Results Processing

To form a statistically stable estimate of the results, a 5-fold cross-validation scheme with 5-fold resampling of the sample (the so-called 5-by-5 scheme) was used. For each method, sets of 25 regression model accuracy estimates were obtained. The obtained

R^2 values were averaged over all runs. We also considered the spread in the quality of the solutions obtained using a statistical estimate of the dispersion of the obtained R^2 values.

The inclusion of various methods for constructing regressors in the study led to the need for top-level equalization of the resources allocated to the method for processing the initial dataset in order to build a regression model. For this purpose, during the preliminary study, measurements of processor time were used, which is necessary for the average computational cycle of setting up the model. The parameters of the approaches used were selected to equalize this indicator. At the same time, the available parameters of the methods were adjusted on each of the tasks to ensure their optimal (taking into account processor time limitations) efficiency on each of the tasks.

3.5 Results and Discussion

ANOVA methods were used to assess the statistical significance of the results. We used the R^2 indicator as a basic indicator for evaluating methods and comparing the effectiveness of the described regression tasks. The research results are presented in Tables 2–4. Analysis of the results obtained leads us to the conclusion that non-ensemble methods are quite effective on relatively simple test problems without noise, since their approximation ability is sufficient. According to our observations, the problems on which such results were obtained include problems numbered 2, 3 and problem Concrete Slump problem. On them, the results obtained by individual solvers and ensemble solvers using several neural network models turned out to be statistically indistinguishable. The results of individual and ensemble solvers on problem 3 are somewhat different, but such a difference is very small and is not a reason for complicating the model in the form of a ensemble.

Among the considered set of test problems, the most complex are the problems of ore-thermal smelting and problems numbered 1 and 4. On such problems, the best results were obtained when using approaches based on ensembles of neural network solvers. Among these ensemble approaches, statistically significantly higher effectiveness corresponds to those that used an ensemble diversity approach. Our assumption is that methods based on single solvers may have reached their efficiency limit in the used configuration. It is the ensemble approach, built even on simpler individual solvers, that makes it possible to overcome this limit when constructing regressors in this case.

Assessing the robustness of the approaches was the subject of further experiments using noisy samples. The maximum noise level was the level indicated below from the value of the test functions used at each point of the definition domain. Table 3 presents the results of the study with a noise level of 10%. As follows from the results, the drop in the quality of approximation for non-ensemble solvers in some cases turned out to be even greater than the level of sample noise. The results of the ensemble solvers turned out to be more suitable for samples with noise simulating those in the measurement channels.

The use of ensemble methods made it possible to achieve more stable results, although a decrease in the calculated indicator is observed. A statistically significant difference was assessed between the method using ensemble diversity maintenance and standard methods for designing neural networks based on evolutionary algorithms.

Table 2. The results of applying methods with no noise.

Approach	Test Problem 1	Test Problem 2	Test Problem 3	Test Problem 4	Test Problem 5	Concrete Slump Data	Ore-thermal Melting Data
Stochastic Gradient Boosting	0.815	0.976	0.934	0.898	0.862	0.984	0.785
MARSplines	0.785	0.981	0.931	0.826	0.773	0.985	0.813
Single Neural Network	0,928	0.968	0.965	0.947	0.851	0.987	0.806
GASEN	0.987	0.996	0.995	0.983	0.902	0.997	0.934
GA-based	0.981	0.997	0.995	0.981	0.901	0.992	0.941
Proposed approach without diversity maintaining	0.983	0.997	0.996	0.983	0.931	0.994	0.961
Proposed approach with diversity maintaining	0.986	0.996	0.996	0.984	0.951	0.995	0.985

Table 3. The results of applying methods with sample noise of 10%.

Approach	Test Problem 1	Test Problem 2	Test Problem 3	Test Problem 4	Test Problem 5
Stochastic Gradient Boosting	0.701	0.911	0.867	0.746	0.776
MARSplines	0.678	0.924	0.854	0.682	0.680
Single Neural Network	0.764	0.917	0.883	0.785	0.757
GASEN	0.931	0.969	0.935	0.923	0.812
GA-based	0.930	0.958	0.942	0.937	0.784
Proposed approach without diversity maintaining	0.939	0.962	0.947	0.937	0.829
Proposed approach with diversity maintaining	0.952	0.987	0.965	0.946	0.880

The complication of the problem associated with the imposition of 20% noise led to an even greater discrepancy in the effectiveness of individual solvers and solvers built on the basis of groups of neural network models. A statistically significant result allows us to

Table 4. The results of applying methods with sample noise of 20%.

Approach	Test Problem 1	Test Problem 2	Test Problem 3	Test Problem 4	Test Problem 5
Stochastic Gradient Boosting	0.637	0.860	0.821	0.639	0.721
MARSplines	0.628	0.851	0.752	0.641	0.626
Single Neural Network	0.681	0.845	0.819	0.695	0.689
GASEN	0.865	0.889	0.891	0.882	0.739
GA-based	0.850	0.905	0.885	0.920	0.721
Proposed approach without diversity maintaining	0.878	0.904	0.912	0.918	0.795
Proposed approach with diversity maintaining	0.927	0.917	0.933	0.936	0.839

provide exactly the proposed approach, with a modification that ensures the diversity of individual regressors in the ensemble. Given that the available computational resources were equalized across different types of regressors, a statistically significant difference in the results of individual and ensemble solvers seems useful in real-world applications. Since the series of experiments used cross-validation schemes trusted for such studies, this allows us to emphasize the ability of the ensemble approach while maintaining diversity to increase the efficiency of solving complex regression problems, the datasets of which are potentially noisy due to objective circumstances. For the problem under consideration with real ore-thermal smelting data, it was also possible to improve the accuracy of the model. An increase in the accuracy of the model for the problem under consideration with real ore-thermal smelting data was also achieved.

4 Conclusion

The study, the results of which are presented in this article, is aimed at creating and evaluating the effectiveness of new methods for constructing ensemble models based on artificial neural networks. Taking into account the high complexity of constructing the structures of artificial neural networks, the study is focused on automating the process of forming an ensemble. A method for forming the structure of neural networks based on a probabilistic evolutionary procedure is considered and described, which makes it possible to automatically generate the structures of neural network models. This author's approach is complemented by new operations aimed at maintaining diversity in the formed ensemble, which is an important factor in ensuring efficiency in the ensemble approach. Despite the fact that direct estimates of the diversity of the models were not carried out (and this is the subject of a detailed statistical study in the future), the conducted numerical study without the use of appropriate operations shows a statistically significant advantage of the diversity maintenance approach. The relative error reduction is 5–7% depending on the problem.

Comparative studies have shown the advantage or equality of the proposed approach in efficiency over alternative evolutionary methods for forming ensembles of neural network regressors. Moreover, this approach requires tuning a smaller number of parameters compared to the standard evolutionary theory used to design neural network structures. This change does not reduce the ability of the approach to adapt to specific problems, as shown during computational experimentation. For a number of problems, the proposed method made it possible to achieve a relative decrease in the error by about 20%. Another advantage of the approach, leading to a reduction in routine computational operations, is the absence of the requirement for encoding and decoding neural network structures into a binary string, which is typical for standard evolutionary heuristics. We carefully used the ANOVA method to check the correctness of the results obtained, which allowed us to establish cases of truly different results in terms of the efficiency of constructing regression solvers. The implementation of the considered approaches in a software system allows us to continue research in the future and implement applications in solving real data analysis problems. In further studies, it is planned to test and adapt the method for efficient use on larger data sets, the processing of which seems to be a resource-intensive problem for ensemble methods.

References

1. Smith, J.L.: Advances in neural networks and potential for their application to steel metallurgy. Mater. Sci. Technol. **36**(17), 1805–1819 (2020)
2. Guo, L., Lei, Y., Li, N., Yan, T., Li, N.: Machinery health indicator construction based on convolutional neural networks considering trend burr. Neurocomputing **292**, 142–150 (2018)
3. Yamashita, R., Nishio, M., Do, R.K.G., Togashi, K.: Convolutional neural networks: an overview and application in radiology. Insights Imaging **9**(4), 611–629 (2018)
4. AL-Qutami, T.A., Ibrahim, R., Ismail, I., Ishak, M.A.: Virtual multiphase flow metering using diverse neural network ensemble and adaptive simulated annealing. Expert Syst. Appl. **93**, 72–85 (2018)
5. Ribeiro, G.T., Mariani, V.C., dos Santos Coelho, L.: Enhanced ensemble structures using wavelet neural networks applied to short-term load forecasting. Eng. Appl. Artif. Intell. **82**, 272–281 (2019)
6. Melin, P., Monica, J.C., Sanchez, D., Castillo, O.: Multiple ensemble neural network models with fuzzy response aggregation for predicting COVID-19 time series: the case of Mexico. Healthcare **8**(2), 181 (2020)
7. Hansen, L.K., Salamon, P.: Neural network ensembles. IEEE Trans. Pattern Anal. Mach. Intell. **12**(10), 993–1001 (1990)
8. Irvine, N., Nugent, C., Zhang, S., Wang, H., Ng, W.W.: Neural network ensembles for sensor-based human activity recognition within smart environments. Sensors **20**(1), 216 (2019)
9. Li, S., Yao, Y., Hu, J., Liu, G., Yao, X., Hu, J.: An ensemble stacked convolutional neural network model for environmental event sound recognition. Appl. Sci. **8**(7), 1152 (2018)
10. Alzubi, J.A., Bharathikannan, B., Tanwar, S., Manikandan, R., Khanna, A., Thaventhiran, C.: Boosted neural network ensemble classification for lung cancer disease diagnosis. Appl. Soft Comput. **80**, 579–591 (2019)
11. Jia, D.W., Wu, Z.Y.: Seismic fragility analysis of RC frame-shear wall structure under multidimensional performance limit state based on ensemble neural network. Eng. Struct. **246**, 112975 (2021)

12. Li, H., Wang, X., Ding, S.: Research and development of neural network ensembles: a survey. Artif. Intell. Rev. **49**(4), 455–479 (2018)
13. Shu, C., Burn, D.H.: Artificial neural network ensembles and their application in pooled flood frequency analysis. Water Resour. Res. **40**(9) (2004)
14. Giacinto, G., Roli, F.: Design of effective neural network ensembles for image classification purposes. Image Vis. Comput. **19**(9–10), 699–707 (2001)
15. Bukhtoyarov, V.V., Semenkina, O.E.: Comprehensive evolutionary approach for neural network ensemble automatic design. In: Proceeding of the 2010 IEEE World Congress on Computational Intelligence, WCCI 2010—2010 IEEE Congress on Evolutionary Computation, Barcelona, Spain, 18–23 July 2010, p. 5586516 (2010)
16. Bukhtoyarov, V.V., Tynchenko, V.S., Nelyub, V.A., Masich, I.S., Borodulin, A.S., Gantimurov, A.P.: A study on a probabilistic method for designing artificial neural networks for the formation of intelligent technology assemblies with high variability. Electronics **12**, 215 (2023). https://doi.org/10.3390/electronics12010215
17. Khwaja, A.S., Anpalagan, A., Naeem, M., Venkatesh, B.: Joint bagged-boosted artificial neural networks: Using ensemble machine learning to improve short-term electricity load forecasting. Electric Power Syst. Res. **179**, 106080 (2020)
18. Liu, L., et al.: Deep neural network ensembles against deception: ensemble diversity, accuracy and robustness. In: 2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), pp. 274–282. IEEE (2019)
19. Ai, S., Chakravorty, A., Rong, C.: Household power demand prediction using evolutionary ensemble neural network pool with multiple network structures. Sensors **19**(3), 721 (2019)
20. Huang, C., Li, M., Wang, D.: Stochastic configuration network ensembles with selective base models. Neural Netw. **137**, 106–118 (2021)
21. Van Roode, S., Ruiz-Aguilar, J.J., González-Enrique, J., Turias, I.J.: An artificial neural network ensemble approach to generate air pollution maps. Environ. Monit. Assess. **191**(12), 1–15 (2019)
22. Ahvanooey, M.T., Li, Q., Wu, M., Wang, S.: A survey of genetic programming and its applications. KSII Trans. Internet Inf. Syst. **13**(4), 1765–1794 (2019)
23. UCI machine learning repository. https://ergodicity.net/2013/07/. Accessed 21 Feb 2023
24. Yeh, I.C.: Modeling slump of concrete with fly ash and superplasticizer. Comput. Concrete Int. J. **5**(6), 559–572 (2008)
25. Gackowski, M., Szewczyk-Golec, K., Pluskota, R., Koba, M., Mądra-Gackowska, K., Woźniak, A.: Application of multivariate adaptive regression splines (MARSplines) for predicting antitumor activity of anthrapyrazole derivatives. Int. J. Mol. Sci. **23**(9), 5132 (2022)
26. Qin, W., Wang, L., Liu, Y., Xu, C.: Energy consumption estimation of the electric bus based on grey wolf optimization algorithm and support vector machine regression. Sustainability **13**(9), 4689 (2021)
27. Friedman, J.H.: Stochastic gradient boosting. Comput. Stat. Data Anal. **38**(4), 367–378 (2002)
28. Lee, M.C., et al.: Computer-aided diagnosis of pulmonary nodules using a two-step approach for feature selection and classifier ensemble construction. Artif. Intell. Med. **50**(1), 43–53 (2010)
29. Yao, C., Dai, Q., Song, G.: Several novel dynamic ensemble selection algorithms for time series prediction. Neural. Process. Lett. **50**(2), 1789–1829 (2019)
30. Koonce, B., Koonce, B.: EfficientNet. Convolutional Neural Netw. Swift Tensorflow: Image Recogn. Dataset Categorization **1**, 109–123 (2021)
31. Van de Ven, G.M., Siegelmann, H.T., Tolias, A.S.: Brain-inspired replay for continual learning with artificial neural networks. Nat. Commun. **11**, 4069 (2020)
32. Mingxing, T., Quoc, V. Le.: EfficientNetV2: smaller models and faster training. In: International Conference on Machine Learning, vol. 139, pp.1–11 (2021)

33. Gaba, S., Budhiraja, I., Kumar, V., Garg, S., Kaddoum, G., Hassan, M.M.: A federated calibration scheme for convolutional neural networks: models, applications and challenges. Comput. Commun. **192**, 144–162 (2022)
34. Raska, S.: Python and Machine Learning. DMK Press, Moscow (2017)